卒業論文

Web ブラウザ上での直感的操作に基づく Web ページ構成法

九州工業大学　工学部　総合システム工学科

09111052

吉田真也

指導教員：　三浦元喜　准教授

平成 25 年 2 月 22 日

概要

現在, SNS やブログなどの普及により個人で Web ページ上へ画像や文章を容易に挿入できるようになった．しかしページの構成をユーザが行うにはある程度の知識が必要で，特に画像を含めるにはアップロード等の作業に手間が掛かる．そこで我々は Web ページ上への画像の挿入をコピー&ペーストで行い，操作のバックグラウンドでアップロードを行うシステムを開発した．本稿ではこのシステムを用いて画像の挿入や編集などを Web ブラウザ上で行い，直感的操作に基づいてユーザが Web ページ構成を行う手法を提案する．本稿では構成の 1 例として手順説明書の Web ブラウザ上での作成支援を取り上げ，システムのユーザビリティを検証するため対照実験を行った．実験の結果，比較対象とした Word と比べ作業の満足度や効率に関して高い評価を得ることができた．これにより我々の提案する手法が，Web ページ構成に対するユーザの手間の解消に貢献できたことが分かった.

目次

第 1 章　導入

本章では，本研究の行う目的，その背景について述べる

1.1　コンセプト

本稿では，Web ブラウザにおけるユーザの直感的操作に基づいた Web ページ構成法を提案する．現在，SNS やブログ等の Web ページにおいてユーザが画像や文章を用いて自由にページを構成できるサイトが多数存在する．しかし Web ページへの画像の挿入には各ページにおけるアップロードのプロセスがあり, それらプロセスの多さが煩わしく問題となっている事がある．これに対しその煩わしさを解消し，より簡単にアップロードを行い Web ページの構成の支援を行う手法が必要であると考えた．そこで我々はコピー&ペーストで Web ページへ画像を貼り付け，同時にアップロードを行うシステムを考案した．本稿では，そのシステムを用いてユーザが画像の挿入などを単純操作で行い Web ページの構成を行うことによる，ユーザビリティの向上を目的とし，以下の章でその手法について詳細に記述する．

1.2　背景

現在，インターネット環境の発展により SNS やブログ等の利用者が増加している．総務省の調査[1]では，国内ブログ利用者が約 335 万人，国内 SNS 参加者が約 111 万人であり，さらに増加傾向にある．これらのサービスでは，文章による表現だけでなく写真や画像をアップロードすることで, そのページの閲覧者と画像を共有することができる機能も備えられている．しかし，画像を Web ページ上で利用する，特に個人のローカルフォルダに保存されている画像を利用する場合, 例えば以下のようなプロセスを経る必要がある．

(1)ページ上にあるファイルのアップロードを選択

(2)ファイルの参照から画像を選択

(3)アップロードを行う

図 1　画像の参照画面の例

このプロセスでは, アップロードを行うに当たっての作業が数ステップに及び手間が掛かってしまう．そこでこれらのプロセスを減らし，より簡単に画像のアップロードを行う手法が開発された．この手法では，コンピュータの操作の 1 つであるコピー&ペースト等によってデスクトップやローカルフォルダに保存されている画像を Web ページ上に貼り付けることができる．これによりユーザが参照画面から画像を選択しアップロードを行うことなく，Web ページへの貼り付けができるようになった．

しかし，これらの技術ではまだ画像を Web ページへ貼り付けた後，サーバへアップロードを行う必要がある．このステップを簡略化するため，Web ページ上へ貼り付けとアップロードを同時に行える機能を探した．そこで，Ajax（後述）という Web ブラウザ内で非同期通信とインターフェースの構築などを行う技術を用いる．これによりページへ貼り付けるという操作を行った際, バックグラウンドでデータをサーバへ送信しアップロードを行うことができる．よってユーザは煩わしいプロセスを経る必要のあっ

た画像のアップロードを，画像をコピー&ペースト等によって Web ページに貼り付けるだけという単一ステップで行うことができる．

ここで我々は，これらの機能を応用して Web ブラウザにおける Web ページ構成が行えないかと考えた．Web ページへの画像の貼り付けを単一ステップで行えるという利点を活かし，更にページ上の任意の位置に貼り付けることができれば，ユーザの好みの Web ページを作成する際の十分な支援になるはずである．そこで我々は Ajax のフレームワークである jQuery による HTML 文書のイベント処理に着目した．jQuery とは JavaScript で記述される Web アプリケーションフレームワークである．これにより，例えばマウスポインタを当てる，ボタンをクリックするなどのユーザが操作したタイミングでコードを実行できる．イベント処理とはこの「ユーザが操作したタイミング」を検出するための仕組みを言う．これによって，より動きを加えたページを作成することができるようになる．我々はこの技術を用いることで，画像を貼り付ける枠をユーザの任意で増やせるように考えた．これにより，ユーザの考える順序で画像を貼り付けることが可能となり，画像を含めた Web ページの構成を行うことができる．

第 2 章　関連研究・技術

今回提案する画像のアップロードおよび Web ページへの構成について，様々な技術，研究を参考とした．本章ではそれらの関連研究について手法ごとに記述する．

2.1　画像のアップロード

Web ページ上での画像の使用において，最も重要となるのが画像の Web サーバへのアップロードである．しかし 2 章で記述した通り，画像を参照画面から選択しアップロードを行うプロセスでは，手間が掛かるという問題点があった．そこで，それらのプロセスをより簡潔に行うことを目的とした技術が開発された．

Joel Besada が開発した Pasteboard[2]は，迅速かつ単純な画像共有 Web アプリである．画像を Web ページ上にペースト若しくはドロップするとページ上に画像が表示され，独自のサーバへアップロードされる．
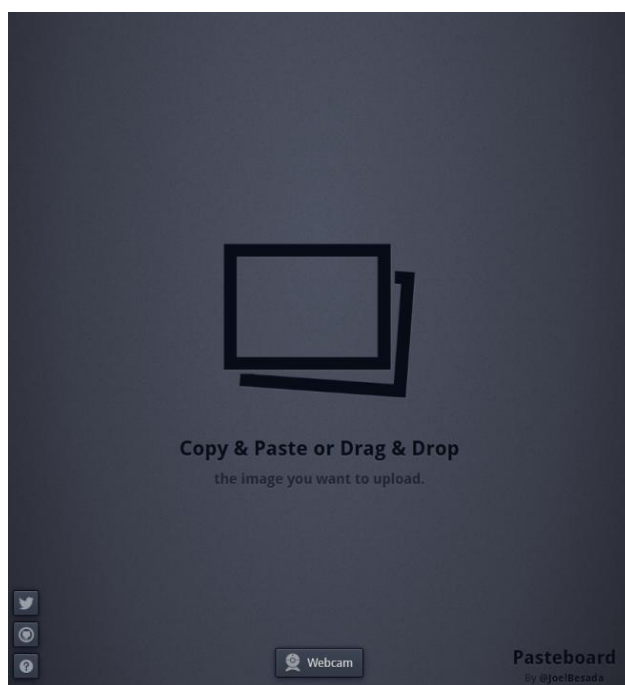
図 2　Pasteboard[2]

また snag.gy[3]では, Photoshop やペイント等の画像エディタでコピーしクリップボードへ保存された画像を単純かつ簡潔にアップロードできる. 更に snag.gy では当サイト上で貼り付けた画像にテキスト入力や切り取りなどの編集を行うこともできる.
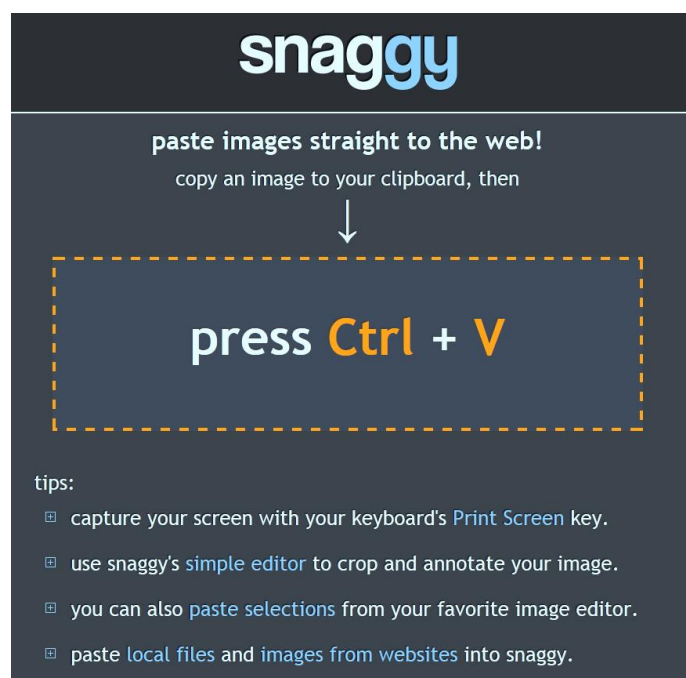
図 3　snag.gy[3]

画像をコピー&ペースト等を行うことなく Web ページへアップロードできる技術も開発された．GYAZO[4]は Web 上から無料でダウンロードできるアプリケーションである．GYAZO を起動するとマウスカーソルが変わり画面イメージを範囲選択できる．ユーザが得たい範囲をドラッグし選択すると，Web ページが開き自動的にアップロードされ，同時にその URL がクリップボードに保存される．

これらの技術では全て，アップロードした画像を Facebook や Twitter などにその場から共有することもできるようになっている．これにより，写真だけでなく Web 上で見つけた画像やユーザが自身で編集した画像も速やかに世界中で共有することが可能となった．

2.2　Web ページの編集

Web ページの作成は，その知識の少ない人には難しく，仮に作ったとしても更新などが面倒となり途中で止めてしまう人がいる．そのような人のために開発されたのがNOTA[5]である．NOTA では，自由自在にデザインを編集することができ，ページの任意の位置に文字や図の挿入や写真の貼り付けを行うことができる．また，Web 上で直接書き込むため作成したページをすぐに公開することができ，グループで共有し複数のパソコンから同時に編集を行うこともできる．特に，NOTA では画面に直接文章を書くことができるため，通常のブログよりも易しく HTML の知識も必要としない．フォントの書式や背景の指定も簡単にでき，手書き入力や図形，写真の配置も自在なため，ページのデザインを自分の好みに応じて編集することができる．

また qwikWeb[6]という Wiki と quickML の統合されたシステムが存在する．このシステムでは，非公開でメール連絡と同時に Wiki を作成できることができ，メーリングリストの利点であるアクション無しで情報を得られる事と，Wiki の利点である情報の蓄積と編集が行える事の両方が備わっている．メールでのやり取りが自動的に履歴として Web ページに蓄積されるため，メーリングリストに参加しているグループメンバーであれば誰にでもページの編集，構成が可能であり，メールのやり取りによるグループの情報共有も容易となっている．

2.3　技術

従来の Web アプリケーションでは，サーバにリクエストを送信した後，その結果を新規の Web ページとして受け取り表示していた．しかし，この方法ではその度に画面が切り替わることで Web の使い勝手が悪くなる，サーバの応答を待つ必要があるなど

の問題が生まれた．そこで Web 関連技術の標準化や高機能なブラウザの普及などを背景に登場したのが Ajax[7]である．Ajax とは「Asynchronous JavaScript + XML」の略で，2005 年アメリカのインフォーメーションアーキテクト Jesse James Garrett により名付けられた．Web ブラウザ内で非同期通信を利用し，その通信結果に応じて動的にページの一部を書き換えインターフェースの構築を行う技術の総称である．これを Web 検索に応用すると，これまで入力確定後に検索を行われていたが，ユーザが入力を行っている間にバックグラウンドで検索を行いリアルタイムに検索結果を表示することができる．これらは Web ブラウザでのみ動作することが特徴で，最も有名な例として Google では Google Map や Gmail などに積極的に Ajax を採用しており，徐々にその実用性が世間に認知され始めている．

本稿では Ajax 用フレームワークとして jQuery[8]を使用している．jQuery とは JavaScript と HTML の相互作用を強化する JavaScript ライブラリであり，HTML 文書のイベント処理やアニメーション等を簡素化し高速で行うことのできるフレームワークである．今回 jQuery を使用した理由は数点ある．まず jQuery がブラウザに依らないオープンソースであること，Web ページ上でのイベント処理の記述を簡潔に行えること，また CSS（Cascading Style Sheets）の操作やエフェクトの指定も行えることなどが挙げられる．

第 3 章　提案手法

本章では我々が提案する Web ページ構成法について詳しく述べる．

3.1　概要

本研究で提案する Web ページ構成法を実現させるため，我々は直感的操作を用いた Web ページ上への画像の挿入，およびアップロードを行い，またページ上において画像の編集が行える機能を加えたシステムを作成した．我々はこれを PaSTePoST システム（以下 P.P.S）と名付けた．この P.P.S は主に以下の

1）画像の貼り付け

2）ページの構成

3）画像の編集

の 3 つの操作で成り立っている．

本稿では提案する手法を用いる簡単な例として，手順説明書の Web ブラウザにおける作成支援を取り上げ，研究を進めた．手順説明書とは，ある目的の動作を行うためにはどういった手順で操作を行う必要があるかを示した文書で，家電などを購入した際に添付される取扱説明書などがこれに当たる．通常，これらの説明書では文章のみでなく画像付きで手順を説明されており，構成も図の順番や文章の配置など作成者が自由に行っている．そのような点を踏まえて，本稿ではこの手順説明書の Web ブラウザにおける作成を軸として提案手法を紹介する．

3.2　直感的操作

本研究では，Web ページの構成において画像の貼り付けやアップロード，編集などを行う際，単純な操作でそれらを可能とする事によりユーザの負担にならないよう考えた．こ

こで単純な操作とは，コンピュータにおける“コピー&ペースト”や“ドラッグ&ドロップ”のようにユーザが感覚的に行える簡単な操作を指す．これらの操作で行う利点は，単純かつ簡単であり，またコンピュータを扱う上で非常によく使う操作であるため，ほぼ全ての人が理解しているという点である．これによりユーザの思うままに操作することが可能であると考え，これをユーザが感覚的に行える操作，即ち直感的操作と呼び我々の提案する手法における基礎とした．

3.3　PaSTePoST システム

本研究では Web ブラウザ上において直感的操作を用いたページ構成を行う手法を提案する．そして，その手法を行うために我々が開発したのが P.P.S である．このシステムにおいてユーザが行う操作は前述の 3 種類を組み合わせたものであり，まず Web ページ上へ画像をコピー＆ペーストにより貼り付け，ユーザの考えるページの全体像に沿って配置し，必要ならば編集を行う，といった流れで行われる．図 4 に P.P.S の Web ページ（以下 P.P.S ページ）の初期画面を示す．ページの上部に画像の貼り付けや編集の操作方法の説明を載せており，その下に説明書に使用する表を作成した．本研究では Web ページ構成の 1 例として手順説明書を取り上げているため，左側に画像枠，右側に文章枠を設けたテーブル構造を採っている．そしてページの状態を保存し再読み込みできるようそれぞれボタンを配置した．また，閲覧ボタンをクリックすると図 5 に示すように操作方法の説明と表の右側についていた＋－のボタンが消され，印刷可能な状態で表示することができる．

P.P.S のプログラム作成で用いた言語は HTML，PHP，CSS および JavaScript である．画像の貼り付けは JavaScript で記述する jQuery のイベント処理や STRd6[15]のサンプルプログラムを参考に作成した．ページの構成には同じく jQuery を利用し，画像の編集ではグラフィックライブラリである GD を使用した．GD とは PHP で JPEG 等の画像処理が行えるライブラリであり, 本研究では画像に関する機能を全て GD によって作成

した．これによりイメージを直接ブラウザに出力することができ画像編集が簡単に行えるように実装できた．

以下に 3 種類の操作についてそれぞれ記述する．

図 4　P.P.S ページ初期画面（作成画面）

図 5　P.P.S ページ閲覧画面

3.3.1　操作 1：画像の貼り付け

本項では P.P.S において基本ともなる画像の Web ページ上への貼り付けの操作について述べる．

P.P.S で Web ページへ画像を貼り付けるには，画像を Ctrl キーと C キーでクリップボードへコピーし，その後 Web ページ上で同様に Ctrl キーと V キーでペーストを行うことにより貼り付けることができる．

まず，画像のコピーに関して述べる．本研究における操作は，デスクトップやユーザフォルダに存在する画像をショートカットキーやマウスの右クリックで表示されるコンテキストメニューを利用しコピーを行うだけなのだが，ここで注意が必要な場合がある．本研究において我々が使用したプログラムでは，画像のビットマップデータをクリップボードより取得しそのデータに基づき，画像の貼り付けや表示を行うという仕様となっている．そのため，クリップボードへコピーされる画像のデータはビットマップデータとなってい

る必要があるのだが，コピーを行う状況によっては画像のデータではなくその場所を示すルートがコピーされる場合がある．その場合に対して本研究ではまだ対処法ができていないため，画像をコピーする場合には，Windows アクセサリのペイント等の画像編集ソフトで画像を開いた状態でコピーを行わなければならない．

続いて画像の貼り付けについては，前述のように画像をコピーした後，ブラウザでの操作となる．貼り付けにおいて重要となることは，マウスオーバーした位置に画像を貼り付けることができる点である．マウスオーバーとはコンピュータの操作画面においてマウスカーソルを対象物の上に重ねると何らかの処理が行われる状態の事である．P.P.S では，マウスオーバーした位置をユーザが視覚的に分かりやすいように，P.P.S ページの画像枠内にマウスカーソルが置かれた時，CSS のプログラムで枠内の背景色が変化するように指定した．マウスオーバー時の例を図 6 に示す．そしてそのマウスオーバーした画像枠内において Ctrl キー+V キーによるペーストが行われると，その操作を読み取り前述の通りにクリップボードへ保存された画像データをページへ呼び出し，貼り付け表示させる．この操作では jQuery のイベント処理によってマウスオーバーされた枠内において貼り付けのキー操作が行われた事を認識させる．そして STRd6 のサンプルプログラムを応用しクリップボードからデータを読み込み，img タグの src 属性値にそのデータを代入しタグ本体を jQuery の append()メソッドでテーブル内に挿入することで Web ページ上に表示される仕様になっている．この時，マウスオーバーされた枠内に既に画像が挿入されているかの判定も行っている．まだ挿入されていない場合はそのままコピーした画像を挿入し，既にされていた場合は一度枠内の要素を削除した後，再度貼り付けの動作を行うように実装している．また P.P.S ページへ貼り付ける画像の拡張子は PNG 形式に指定している．これは Web ページ上で画像の編集を行う際に画質をあまり落とさずに行いたいと考え，PNG 形式の拡張子を選択した．

画像付きの説明書において，画像を残すための保存は不可欠な操作である．そこで P.P.S

では画像の貼り付けと同じタイミングで保存，即ちサーバへのアップロードを行うようにした．具体的には，画像が Web ページ上に貼り付けられた時，その画像データを構成するバイナリデータを base64 というエンコード方式を用いてテキストデータへ変換し，そのデータを随時名前を付けてフォルダへ送信，保存するという形となっている．しかし，通常プログラムは書かれてある事をその順番に沿って実行するため，同時に行うことは難しいと考えていた．そこで我々はユーザの貼り付け操作のバックグラウンドで保存を行わせようと考え，Ajax の技術を用いてサーバ側で処理を行うように指定することでこの問題を解決した．具体的には，前述のように画像の挿入がプログラム上で行われた後，ajax()メソッドにて保存方法などを含めた PHP ファイルを非同期通信で実行する．PHP では Image 関数を用いて，まずコピーした画像のデータを読み込んで新たに画像を作成し，名前を付け指定したフォルダに出力することで画像が保存される．今回これらの処理が行われた日時を画像の名前に設定することで，名前が重複することなく貼り付ける度に何枚でも保存できるようにプログラミングを行った．これによって，これまで別操作で行っていた保存を，貼り付けと同時という 1 ステップで行うことに成功し，ユーザの手間という面を大きく減らす事が出来た．

このように，P.P.S では画像をクリップボードへコピーし，そのデータをブラウザ内のマウスオーバー位置を認識して Ctrl キー+V キーによる貼り付けを行い表示する．そしてそのデータを保存しやすいように処理を行ってユーザの操作のバックグラウンドで保存を行うことができる．これが，画像の貼り付けについて行われる動作となる．次項では，貼り付け後の Web ページの構成方法について述べる．

図６　マウスオーバーした状態
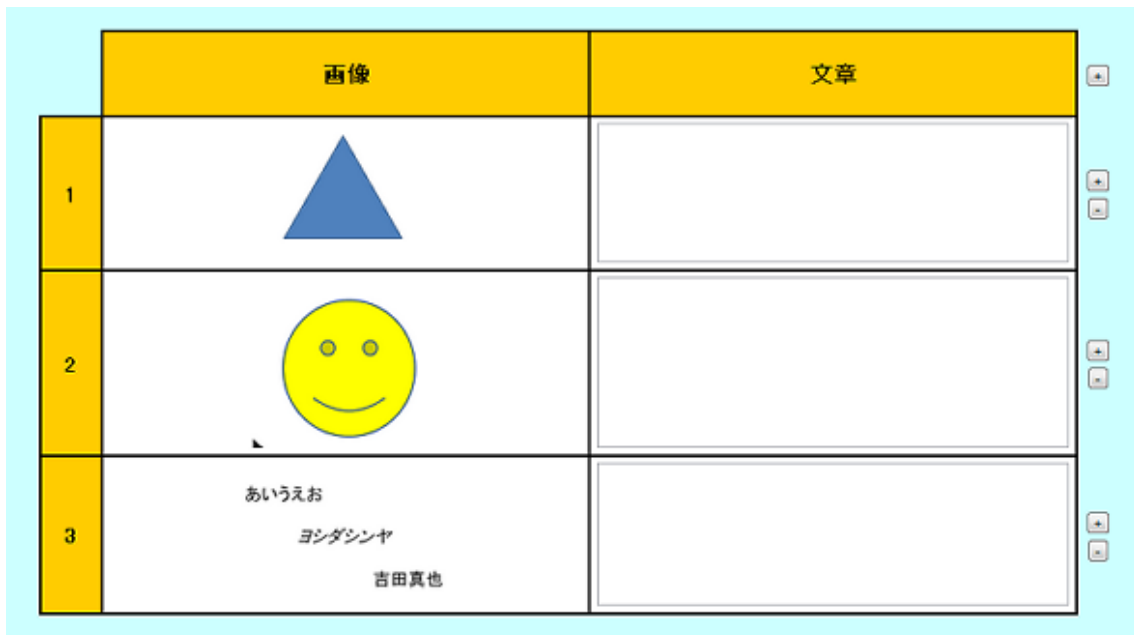
図７　画像の貼り付け後

3.3.2　操作 2：ページの構成

本項では Web ページの構成について述べる．本研究では，Web 上における手順説明書の作成を例として取り上げプログラムの作成やレイアウトを行った．そのため，手順の数に合わせて，テーブルの枠を追加または削除することでページの構成をユーザが変更できるようにと考えた．テーブルの右隅に＋ボタン，－ボタンがある．機能は＋ボタンがテーブル行の追加であり，ボタンがクリックされたという処理を認識するとテーブルの最下行の下に同じ行を作成するという仕様になっている．この機能には HTML の onclick 属性を使用している．これはボタンがクリックされたことを認識したときに行う動作を指定でき，今回は認識した時 JavaScript ファイルの構文を実行するように実装した．＋ボタンが押された時，テーブルの子要素の最後に append()メソッドで行全体のタグを挿入する．これにより仮に手順が予想より多くなってしまった場合にも，行を追加することで対処することができる．また－ボタンが削除の機能を持ち，これは＋ボタンと同じく onclick 属性で動作を指定し remove()メソッドによって行を削除する，作成中に必要の無くなった行ができた場合には－ボタンをクリックすることでその行を消すことができる．
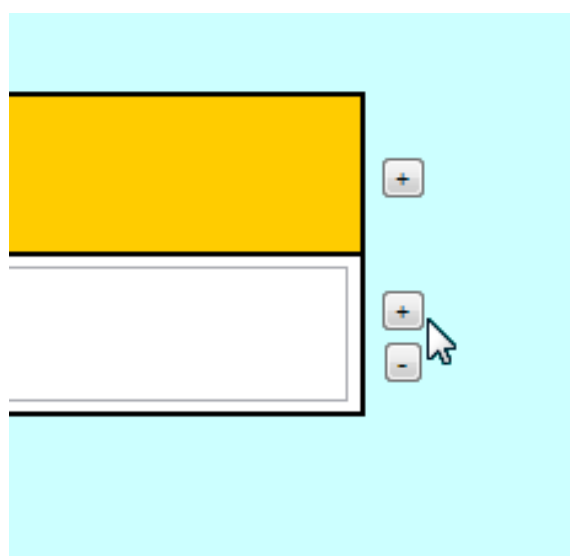
図 8　行の追加／削除ボタン
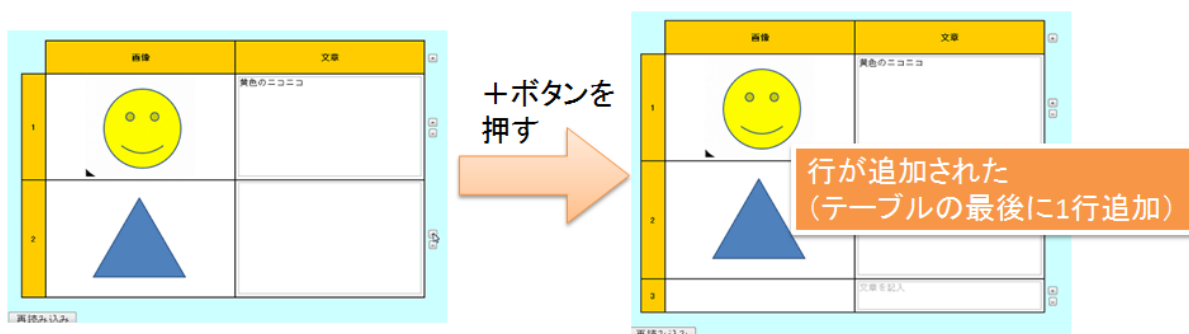
図 9　行の追加
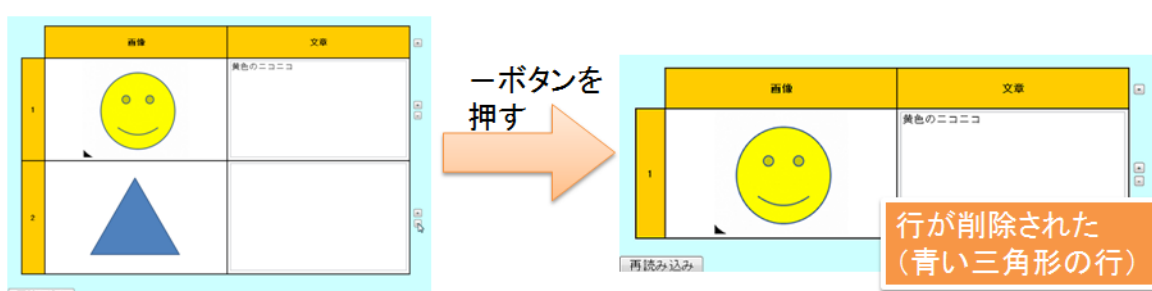
図 10　行の削除

3.3.3　操作 3：画像の編集

本項では，Web ページ上へ貼り付けた画像に編集を行う方法について記述する．3.3.2 までに記述した操作によって画像を Web ページ上へ貼り付け，またその枚数もユーザによって自由に変更できるようになった．しかし，画像をコピーするだけでは何が重要でどこを主張しているのかが閲覧者には理解できない．そこで我々は P.P.S に Web ページ上で画像の編集を行える機能を追加した．操作方法は後述に詳しくまとめるが，大まかには画像上でドラッグしキー操作を行う．ここでどのキーが押されたかの認識には，JavaScript の keyCode プロパティによってキーの ASCII コードを取得し，if 文を用いて判定を行っている．さらに編集した画像もその操作の度に 1 枚の画像として保存されるようにし，後で途中の状態の画像を確認する事もできるようになっている．保存の際どのような編集を行った画像なのかが分かるように，画像の名前にそれぞれキー操作と同じアルファベットを付加した．編集方法には矩形の描画や切り取りなど数種類を

用意し，それぞれのプログラムは PHP で GD を用いた実装を行うことで Web ブラウザに編集した結果を直接表示できるようになっている. 以下でそれぞれの編集方法について詳しく説明する.

・矩形描画（Rectangle）

　手順説明書において, 画像中に強調したい箇所や示したい部分が存在する場合がある. P.P.S ではそのような場合に対し, 画像中に矩形, 即ち四角形を描く機能を加えている. 画像上でマウスをドラッグし，描画したい部分を選択，マウスを離すか範囲選択中に R キーを押すと矩形が画像に描かれる. これには imagerectangle()関数を用い座標は始めにクリックした位置とマウスを離した位置を識別しパラメータとして代入している. これにより, 文章で説明されている部分が画像のどの部分に相当するのか閲覧者に視覚的に分かるようになっている.

　図 11 を例に操作を説明する．画像中のスマイルの部分を矩形で囲みたいとする．まずスマイルの周りをドラッグで選択する．その後マウスを離すか，R キーを押すと確認のウィザードが出るので，そこで OK を押す．これによりスマイルを矩形で囲むことができる.
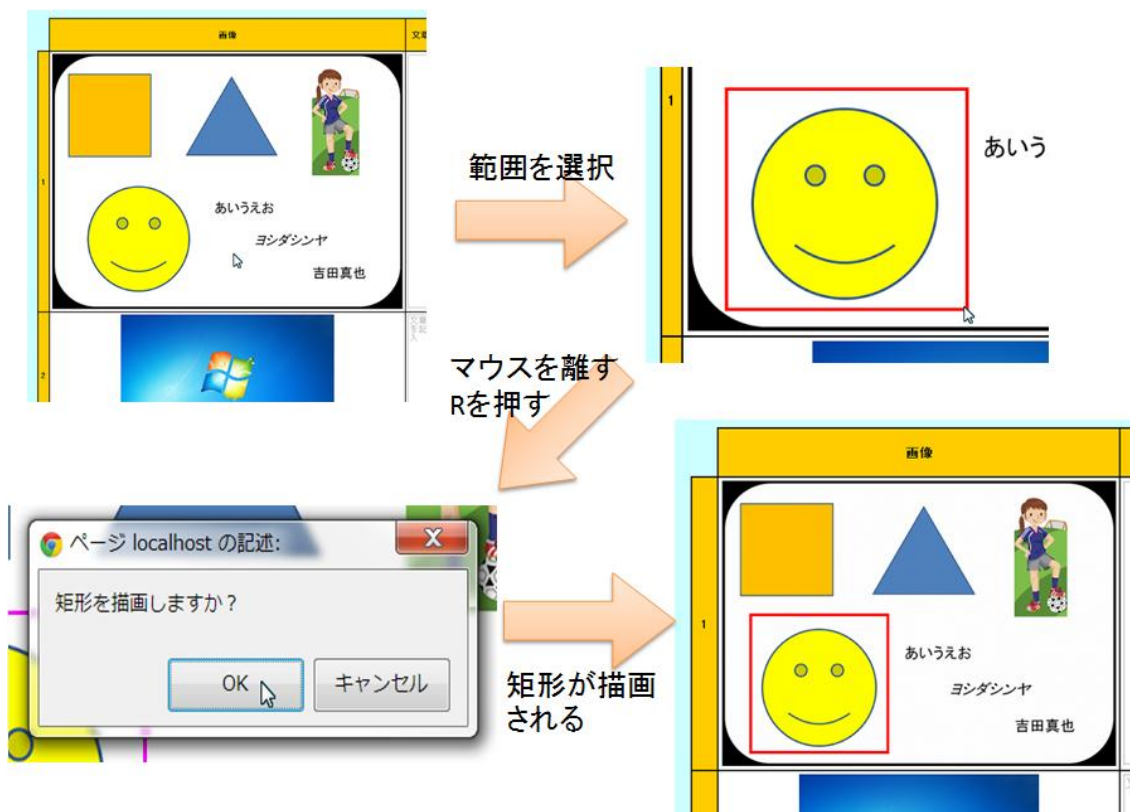
図 11　矩形描画

・切り取り（Crop）

画像の編集においてもっとも多用するだろう機能が切り取りである. 切り取りを行うと, 示したい部分のみを表示することができるため編集前に比べ強調したい部分がより分かりやすくなるという利点がある.

P.P.S では画像上でマウスをドラッグすると四角で範囲を選択でき，切り取りたい部分を囲んだ状態で C キーを押すとその部分が切り取られ表示される．ここで注意が必要なのだが, 選択中にマウスから指を外すと矩形描画の操作となってしまうため切り取りや以下の編集操作を行う際には, マウスを離さずにキー操作を行わなければならない. 操作は切り取りだが，プログラムではドラッグして選択した範囲を imagecopy()関数でコピーし差し換えるという形を取っている．これにより，Web ページ上では画像が切り取られたように表示することができる.

図 12 に切り取りの例として，青い三角形の切り取りを示す．まず三角形をマウスのドラッグで範囲選択し，その状態でマウスから指を離さずに C キーを押す．すると範囲で選択された部分が切り取られ，選択部分に含まれた青い三角形が表示される．
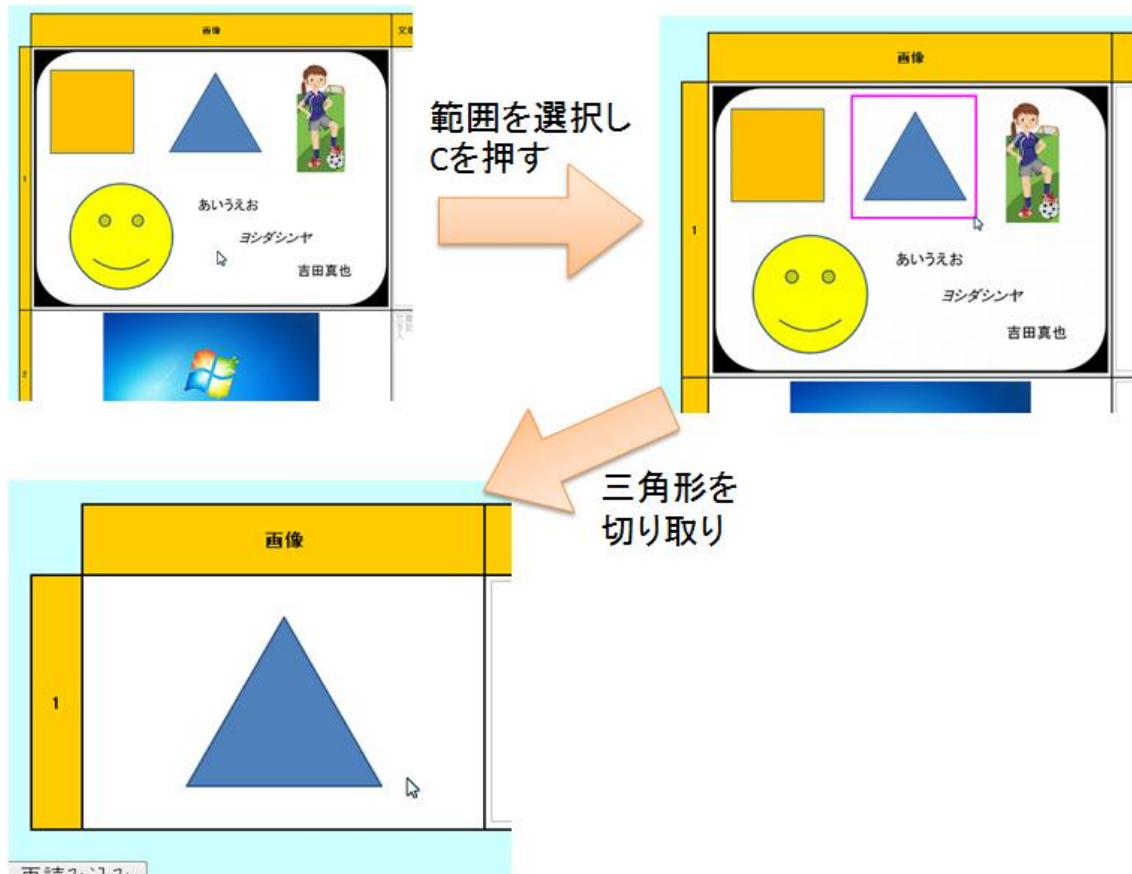
図 12　切り取り

・モザイク化（Mosaic，Pixelize）

次にモザイク化について説明する．モザイク化とは例えば画像によって見えない方がよい部分があった場合などにその部分を粗く表示しぼかすことで内容を隠す方法である．これにより，切り取ることはできないが説明上必要のない部分や隠したいところに対しモザイク加工を施すことで，閲覧者への配慮を含めた画像を作成することができる．

加工方法は，切り取りと同様にモザイクをかけたい部分をドラッグで選択し M キーまたは P キーを押す．すると選択した部分のみが粗く表示されモザイク化される．こ

れはまず imagecopy()関数でドラッグした部分をコピーして新しい画像として生成し，その画像に imagefilter()関数の IMAGE_FILTER_PIXELATE タイプでモザイク効果を適用する．そして処理を行った画像を再び元の画像のドラッグで取得した座標にコピーすることで，選択した部分がモザイク化された画像となる．更に繰り返し同じ処理を行うことでより粗い表示にすることもできる．

図 13 にモザイク化の例を示す．カタカナのテキスト部分をモザイク化し見えにくくしたいとすると，まずマウスドラッグでカタカナを範囲選択する．そしてそのまま M キーまたは P キーを押すと選択された部分が全体的に粗くなりモザイク化される．更に繰り返し同じ箇所にモザイク加工を施すことでほぼ見えない状態にすることができる．
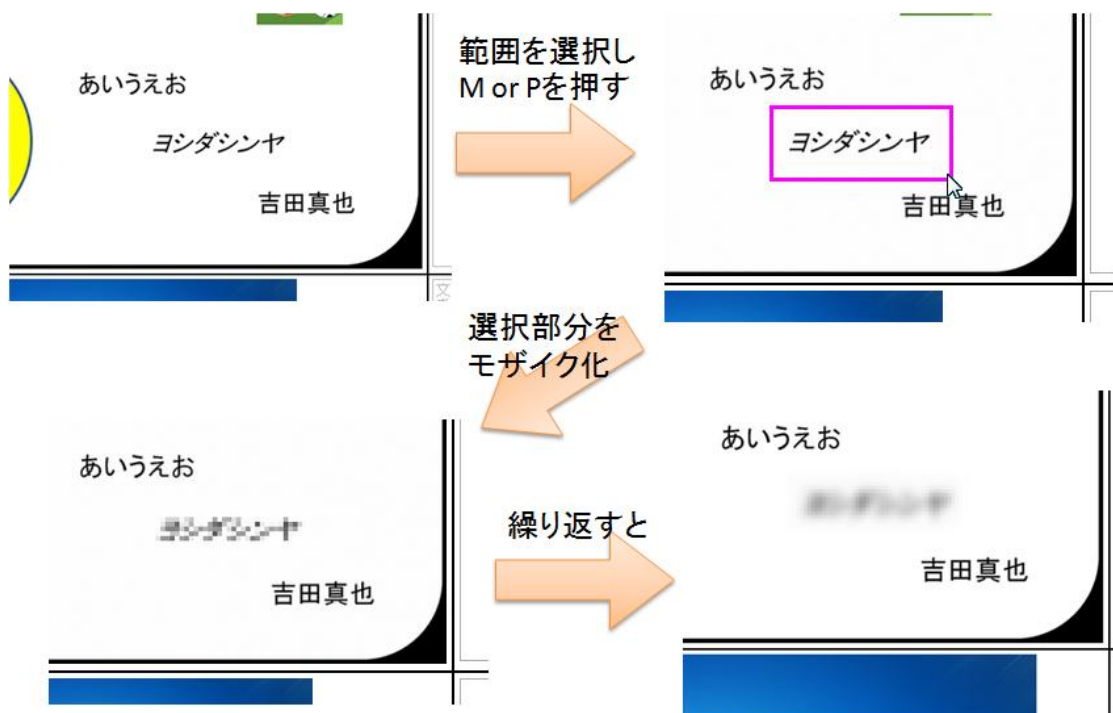
図 13　モザイク化

・縮小（Small，Half）

画像をコピーし Web ページへ貼り付けた際，画像のサイズが大きすぎるため見にくくなる場合がある．そのような場合に対処するためには，画像を縮小する必要がある．P.P.S では元の画像に対し 25％と 50％の倍率で縮小できるようにし，大きいサイズの画像を貼り付けた後でも見やすい状態へ変更できるように機能を加えた．操作方法は，まずマウスをドラッグしその状態で S キーまたは H キーを押す．この操作ではまず imagesx()関数および imagesy()関数で画像の幅と高さを取得する．この値を元に S キーを押した場合は取得した値を 3/4 倍し imagecopyresampled()で再度サンプリングを行い表示する．H キーを押した場合はこの処理を 1/2 倍した値で行う．これにより S キーを押すと 25％縮小され，H キーを押すと 50％縮小することができる．ここで，前述の切り取りやモザイク化では範囲を指定していたが，縮小や後述する削除，やり直しの編集方法では範囲を指定する必要がなく任意の大きさでドラッグし各キーを押すことで画像の編集を行うことができる．

図 14 において画像の縮小の例を示す．縮小したい画像の上でマウスをドラッグし S キーを押すと元の画像に対し 25％縮小された画像となる．また H キーを押すと 50％縮小された画像となり，サイズの大きい画像に対し全体が見やすい大きさに変更することができる．
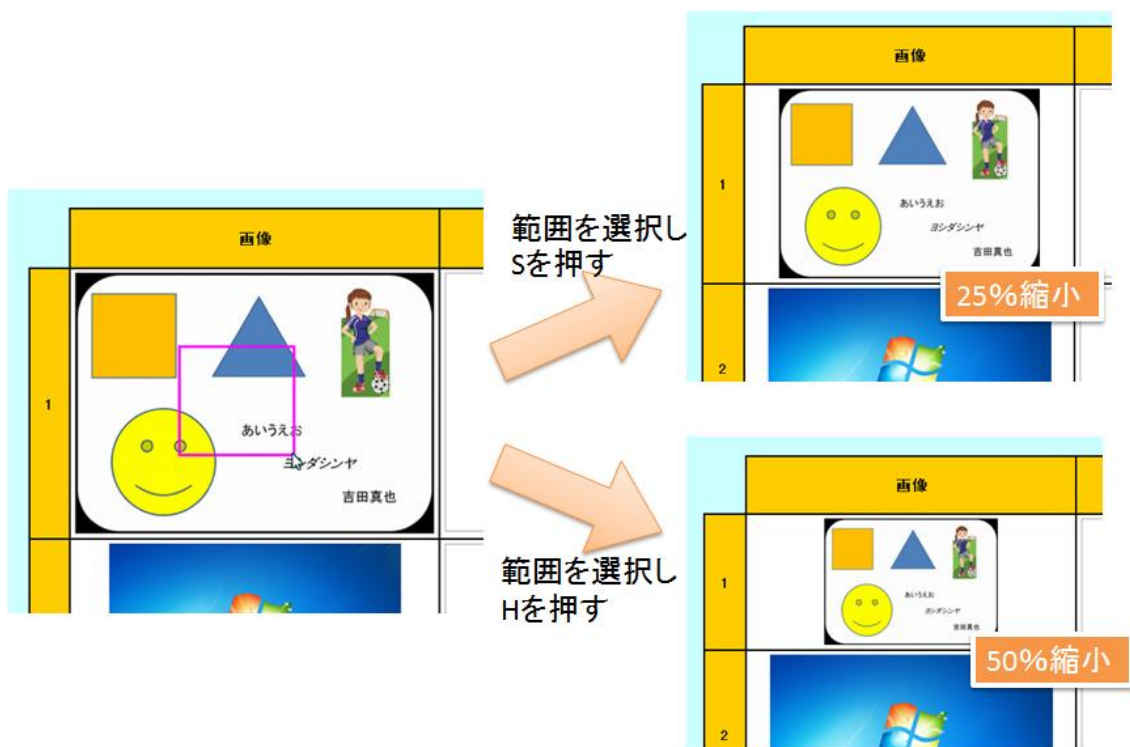
図 14　縮小

・削除（Delete）

説明書の作成において，一度 Web ページへ貼り付けた画像よりも適したものを見つけ貼り直したい場合や，コピーする画像を間違えて貼り付けてしまう場合がある．このような時，画像を削除または差し換えができると便利である．P.P.S ではそのような時にも対処すべく機能を追加した．

画像の差し換え方法は簡単である．まず新しく差し換える方の画像をコピーしクリップボードへ保存する．次に通常の貼り付け操作と同様に Web ページ上の間違えた画像にカーソルを置く．そして Ctrl キー+V キーによって画像を貼り付けると新しい画像に差し替えることができる．例を図 15 に示す．P.P.S ページで表示されている青い三角形の画像を黄色のスマイルに変えたいとすると，まず差し換える黄色のスマイルの画像をクリップボードへコピーする．そして三角形の上へカーソルを置き Ctrl キー+V キーでペーストを行うと，三角形の画像をスマイルへと差し換えることができる．

続いて削除の方法を説明する．削除については，削除したい画像上でマウスをドラッグし，その状態を維持した上で D キーを押す．すると「削除してよろしいですか？」という確認のウィンドウが開くので，そこで OK をクリックするとその画像を削除することができる．削除のプログラムは他の編集操作と異なり PHP ではなく JavaScript で記述している．D キーが押されたことを認識し empty()メソッドで要素を削除することで画像だけを除いている．図 16 は 1 行目の画像を削除する操作の例を示している．削除したい画像の上でドラッグし，そのまま D キーを押す．すると右上のように確認のウィンドウが表示され，ここで OK をクリックすると 1 行目にあった画像を削除することができる．
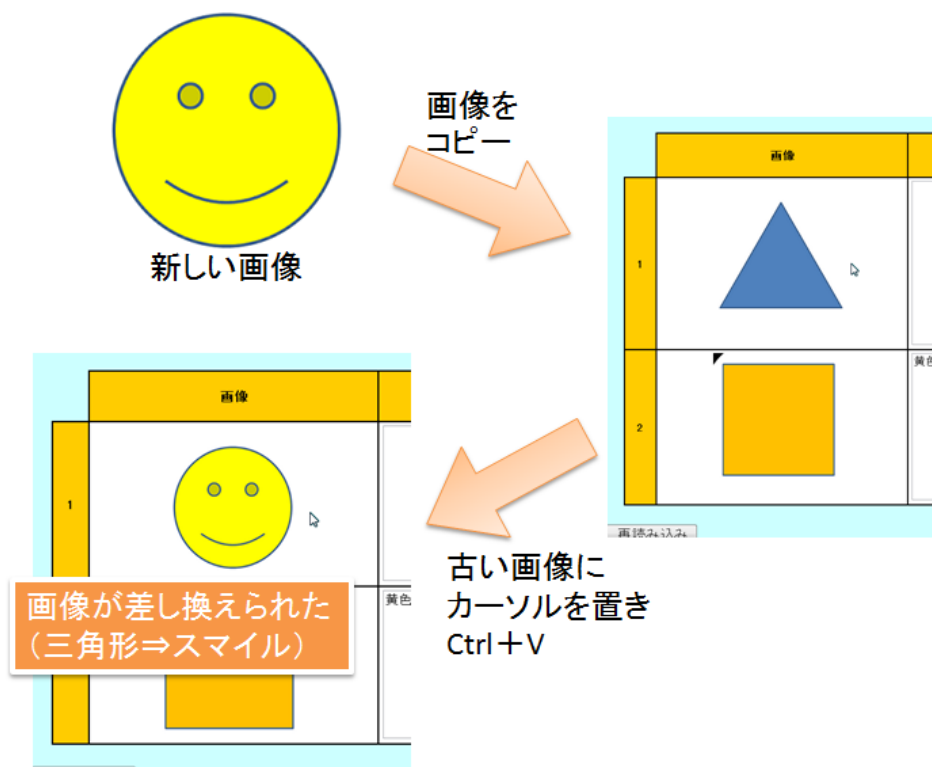
図 15　差し換え
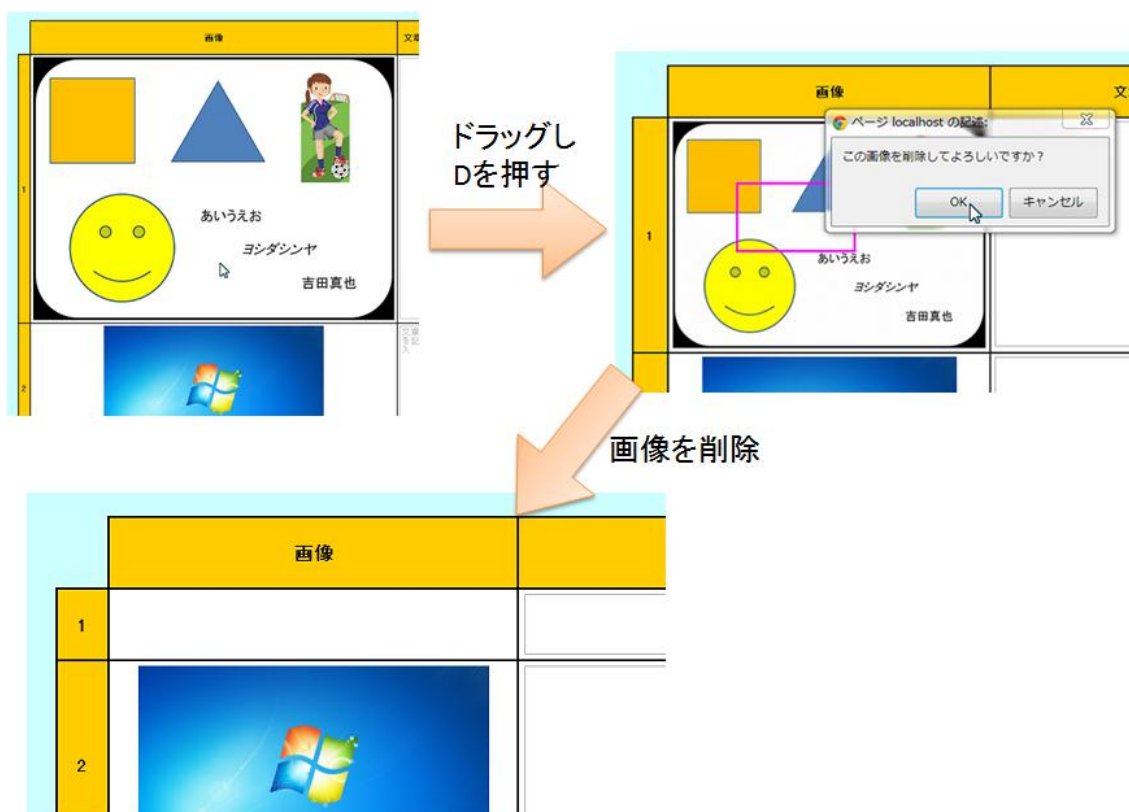
図 16　削除

・やり直し（Undo，Original）

画像の編集中，矩形描画の範囲を間違えたり，D キーを押して削除を行うはずが誤って C キーを押して切り取りをしてしまったりなど，ユーザが編集ミスを犯すことがある．そのような場合，画像を差し換えて新たにやり直すのは面倒が増え時間も掛かってしまう．そのため，編集において操作をやり直す，元に戻すといった機能は必要不可欠である．P.P.S では 1 つ前の編集操作を止める，即ち元に戻す操作（Undo）と，編集前の最初に貼り付けた状態に戻る操作（Original）の 2 種類を備えている．それぞれのキー操作は，Z キーで元に戻る操作，O キーで最初に戻る操作としている．Undo の操作で Z キーを割り当てたのは，通常コンピュータを扱う上でこの操作を行う際に慣習的に Ctrl キー＋Z キーを用いている事が多いため，今回このキーを Undo に割り当てた．プログラムに則って説明すると，元に戻す操作では 1 つ前の操作を行った時の画像を呼び

出し，元あった画像を削除することで操作をやり直せるようにしている．即ち 1 つ前の操作で保存した画像と今の画像とを差し換えるような操作となっている．また最初に戻る操作では，同様に初めに貼り付けた時の画像を呼び出し今の画像を削除して表示させている．これにより，仮にユーザの編集ミスで画像が意図しないものになってしまった場合にも，やり直しの操作を行うことでミスをする前の状態へ戻ることができ，試行錯誤を繰り返してより良い編集を行うことができるようになる．

図 17 にやり直しの操作の例を示す．図中の右上への矢印が元に戻す操作である．左の元の画像において最後の編集操作でひらがなの“あいうえお”に矩形を描いたが，これをやり直し元に戻したいとする．このような場合は，元の画像の上でドラッグし Z キーを押すと最後の操作を消した，矩形が外れた画像となる．また，複数回編集を行った左の画像の上でドラッグし O キーを押すと，図中の右下のように初めに貼り付けた画像の状態に戻る．
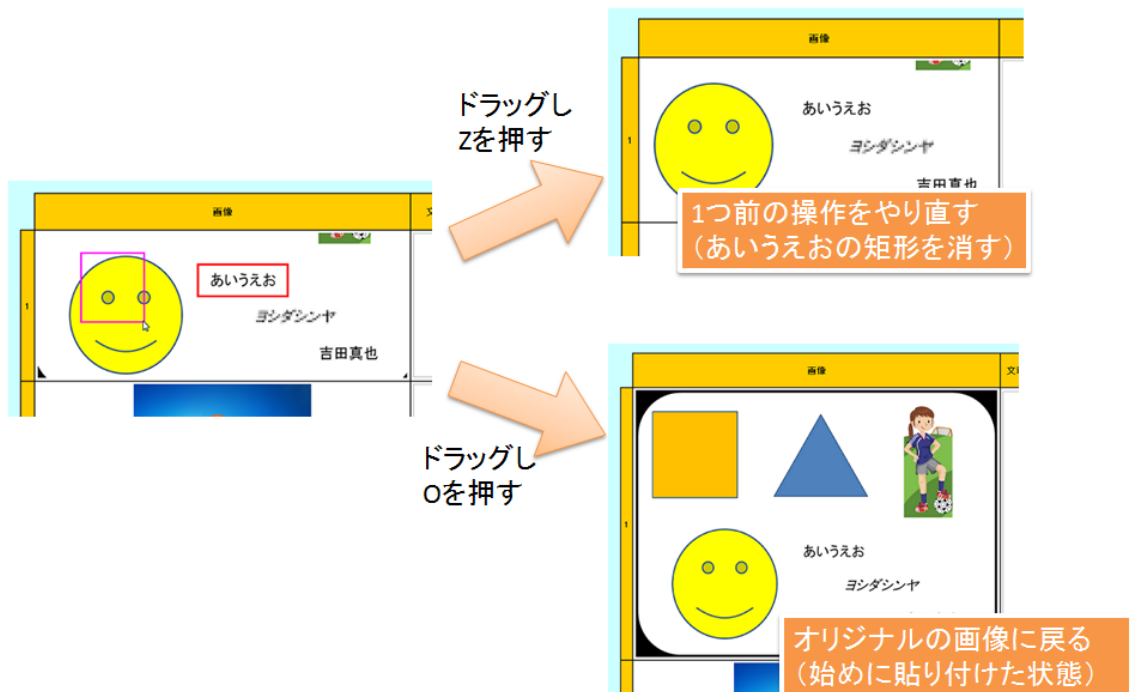
図 17　やり直し

第 4 章　実験・検証

本研究では P.P.S によるシステムのユーザビリティの向上を検証するための実験を行った．本章ではその実験の手順および結果と考察について述べる．

4.1　概要

4.1.1　実験内容

本実験では，提案する P.P.S による Web ページへの画像の挿入に対する手間の解消や編集に対するユーザビリティを評価し，P.P.S によって Web ページの構成がどの程度簡略化でき，ユーザの負担にならない作業ができるかを検証する．前述のとおり本研究では Web ページ構成法の例として手順説明書を挙げている．よって本実験においても，画像付きの手順説明書（以下説明書）を 2 つの方法で作成し評価する対照実験を行う．方法 1 は画像を別ソフトで作成，保存し，Microsoft Office Word 2007 を用いて説明書を作成する．方法 2 では P.P.S を用いて画像を貼り付け Web ページ上で編集を行い作成する．説明書は通常，画像と文章とで構成されているが今回は被験者の負担を減らすため文章はこちらで用意し，文章に対応した画像を作成するという作業を被験者に指示する．これら 2 つの方法に対し 2 種類の手順説明書を作成し，それぞれについて 5 段階の評価と自由記述をアンケートに採る．そしてそれらの結果を元に考察を行う．

4.1.2　使用するもの・環境

今回の実験では，事前に被験者がどの程度 Word 及び Web ページに関する知識や画像編集に関連した経験があるかを知るため，被験者個人のスキルを問う調査票を用意した．以下に調査票に記載した項目を記す．

□　Word の起動はできますか？

□　Word の終了はできますか？

□　タブとはどれか，またその機能はわかりますか？

□　リボンとはどれか，またその機能はわかりますか？

□　ツールバーとはどれか，またその機能はわかりますか？

□　ファイルやフォルダという言葉の意味がわかりますか？

□　Word の文書を保存したことがありますか？

□　Word の文書をファイルの形式を変えて保存した事がありますか？

□　範囲を選択して文字列や行のコピー・挿入・削除をしたことがありますか？

□　文字の書体（フォント）・大きさ（フォントサイズ）を変えたことがありますか

□　太字・斜体や下線（アンダーライン）を使用したことがありますか？

□　文書の中に表を作成（挿入）したことがありますか？

□　セル・行・列の言葉の意味がわかりますか？

□　表の中に行や列を（追加）挿入したことがありますか？

□　表の中から行や列を削除したことがありますか？

□　行の高さ・列の幅を変更したことがありますか？

□　文書に画像を挿入したことがありますか？

□　文書に貼り付けた各種画像のサイズを変更したことがありますか？

□　Web ページも印刷物も画像とテキスト（文書）で構成されていることを知っていますか？

□　Web ページ内の画像を保存したことがありますか？

□　Web ページ内で文章を入力したことがありますか？

□　Web ページ内に含まれるボタンを操作したことがありますか？

□　Web ブラウザの基本操作（例：更新，戻る）を行ったことがありますか？

□　Web のフォームを利用して画像をアップロードしたことはありますか？

□　ブログを利用したことがありますか？

□　Facebook などの SNS を利用したことがありますか？

□　SNS やブログ等に画像をアップロードしたことがありますか？

□　画像ファイルの読み込み（キャプチャ）をしたことがありますか？

□　ビューワと呼ばれる画像表示ソフトについて知っていますか？

□　パソコンに標準的に導入されている画像編集ソフトについて知っていますか

□　画像編集ソフトで画像の編集を行った事がありますか？

□　画像の（縦横）サイズとは何か分かりますか？

□　画像のサイズを変える方法を知っていますか？

□　画像サイズをあらわすピクセルとは何か分かりますか？

□　画像のトリミングとは何か分かりますか？

□　画像をトリミングする方法を知っていますか？

□　クリック，ダブルクリック，右クリックの意味・活用方法を知っていますか？

□　Ctrl キーや Shift キーを用いたショートカットキーの操作をしたことがありますか？

□　ファイルのコピーと切り取り（カット）の違いを知っていますか？

□　クリップボードとは何か，その機能を知っていますか？

□　PrintScreen キーとは何か，その機能を知っていますか？

□　PrintScreen キーを使用した事がありますか？

□　スタートメニューとは何か知っていますか？

□　アクティブウィンドウとは何か知っていますか？

□　コントロールパネルとは何か知っていますか？

□　デスクトップ（画面）とは何かわかりますか？

□　タイムゾーンとは何か知っていますか？

□　スクリーンセーバーとは何か知っていますか？

以上.

本実験では，被験者への公平性を保つため同一のコンピュータを利用する．パソコンのバージョンはWindows7を使用し，方法1ではMicrosoft Office Word 2007を用いて文書を作成させ，画像の取り込みはキャプラ[9]（図18）というソフトをセットで用いる．キャプラとはウィンドウ等の画面取り込みを行うフリーのキャプチャソフトである．これを用いることでスクリーンショットやアクティブウィンドウの取り込みなどを簡単に行え，更にキャプチャした画像のサイズ変更や一部の切り抜きなどの簡単な編集も行うことができるようになっている．
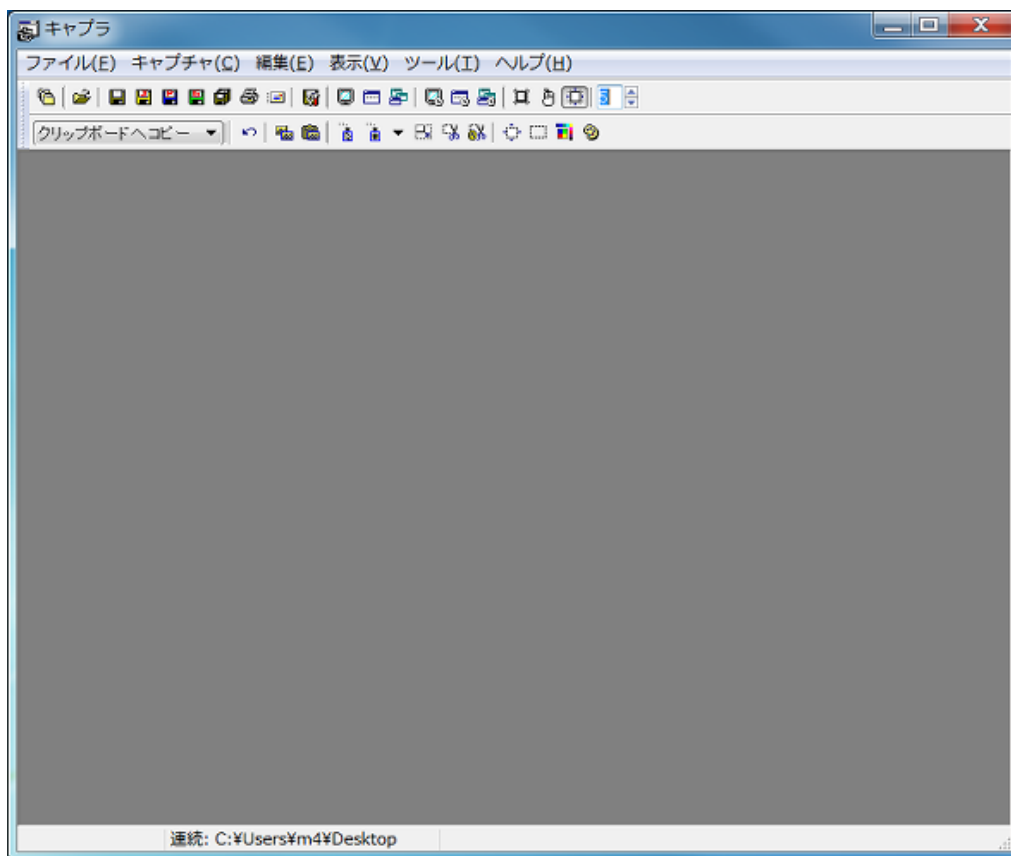
図18　キャプラ画面

方法 2 の P.P.S は XAMPP[10]を用いてローカルネットワーク内でプログラムを作成し，Google Chrome[11]で動作させた．画像の取り込みはキーボードの PrintScreen キーを用い，編集も Web ページ上で行える操作に限定した．

4.1.3　被験者

本実験では，被験者は 12 名で，全員が九州工業大学工学部所属の学生である．事前に記載させた調査票によるとスキルに大差はなく，全員に対し平等な実験を行うことができた．また全員以前に本実験と同様の手順説明書を作成した経験は無く，その面でも十分に公平な実験となった．

4.2　実験方法

本実験では被験者 1 人に対し，用意した 2 種類の手順説明書を 2 つの方法でそれぞれ別に作成するため全体で 4 通りの進め方がある．具体的には Word と P.P.S の方法で手順 1 と手順 2 があるため，その組み合わせは

（1）Word で手順 1　⇒　P.P.S で手順 2

（2）Word で手順 2　⇒　P.P.S で手順 1

（3）P.P.S で手順 1　⇒　Word で手順 2

（4）P.P.S で手順 2　⇒　Word で手順 1

となる．

実験では，まず 1 つ目の方法で 1 つ目の説明書を作成してもらい，それに対する評価を行う．その後 2 つ目の方法でもう 1 つの説明書を作成し評価をする，といった流れで行った．実際の作業については，Word ではキャプラの，P.P.S ではページ上での編集操作に慣れてもらうため，一度リハーサルを行いその後同じ作業をさせた．これは，通

常説明書を作成する場合，その手順に対しても作成する方法に対しても十分に知識がある状態で行うと考えられるため，ある程度慣れた状態で作成することが望ましいと考えたからである．

4.2.1　方法 1：Word＋キャプラによる作成

本項では Word とキャプラを用いた説明書作成の方法を述べる．大まかには，事前に用意した文章に沿った画像を作成し Word で作成した説明書に挿入し保存を行う，といった流れになっている．

（1）用意された文章を読み，それに沿った画面を表示する．

（2）キャプラを用いてその画面のスクリーンショットを取り，サイズ変更や切り抜き，またはマウスカーソルの埋め込みなどの編集を行う．

（3）画像は任意のフォルダに保存し，画像そのものを参照できるように残しておく．

（4）Word へ移動し，事前に作成した表の画像枠内に保存した画像を挿入する．

（5）全ての手順について『キャプチャ→編集→保存→挿入』を行う．

（6）終了したら文書を PDF 形式で保存する．
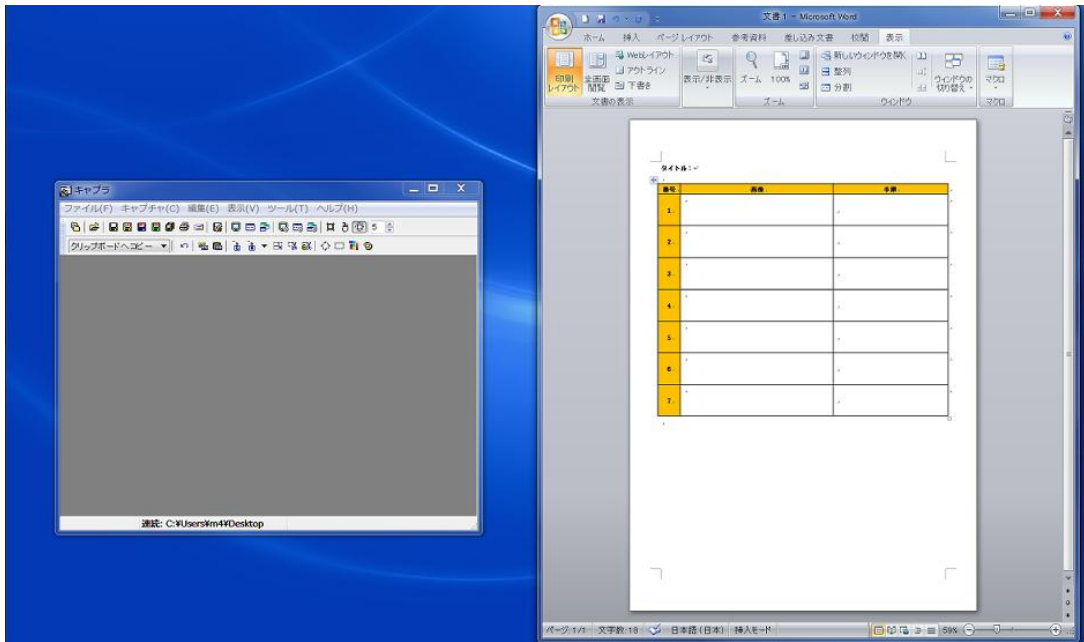
図 19　方法 1 操作画面（左：キャプラ　右：Word）
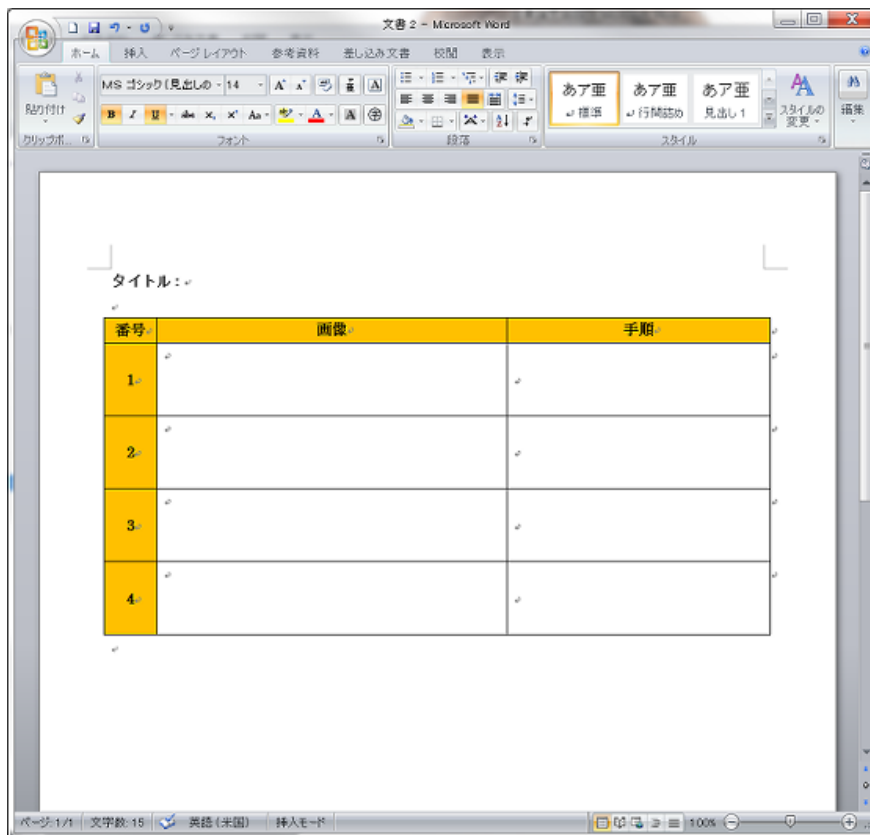
図 20　Word の説明書フォーマット（左：画像枠　右：文章枠）

4.2.2　方法 2：P.P.S による作成

方法 2 では前述の Word と同様の作業を我々が提案する P.P.S を使用して行う．こちらも事前に用意した文章に沿って画面をキャプチャし編集を行い，説明書を完成させる．方法 1 との違いはキャプチャした画像を別ソフトを介せずに Web ページ上に挿入することができ，編集もページ内で行えることである．

（1）事前に用意された文章を読み，それに適当な画面を表示する．

（2）画面をキャプチャしクリップボードへコピーする．

（3）P.P.S ページの画像枠内にマウスカーソルを置き Ctrl キー＋V キーで貼り付ける．（この時サーバ側で画像の保存が実行される）

（4）ページ上部の操作方法に従って，画像の編集を行う．

（5）全ての手順に対して『キャプチャ→コピー→貼り付け→編集』を行う．

（6）終了したら保存ボタンを押し閲覧画面へ移行する．

4.3　アンケート

本実験では手順説明書の作成の際の画像挿入や編集におけるユーザビリティを検証することを目的とした．そこで実験後に行う被験者へのアンケートについてもユーザビリティの定義[12]に則った項目を用意した．ただし，ここで参照した ISO9241-11 におけるユーザビリティの「有効さ」である“目標を達成する上での正確さ，完全性”については，Word と P.P.S の実験では共に備わっていると考えアンケートの設問には関連するものを設定していない．Word および P.P.S について，作業後の完成した文書や作業自体に対する満足度，作業の効率などについて同じ内容の設問を用意し比較しやすいよう設定した．また，2 つの方法を場合によってどのように使い分けるかを聞き，利用状況へのユーザの感想も同時に聞く事にしている．以下が Word および P.P.S に対する

アンケートの項目であり，【　】はその項目で被験者へ問いたい内容を示す.

1. Word についてお聞きします

・完成した文書はあなたにとって満足のできる仕上がりになりましたか【作品の満足度】

・Word では画像を挿入した文書を作り易いと思いますか【効率】

・作業は心地よく（不快感なく）できましたか【作業の満足度】

・今後同様の作業をする場合，Word をどの程度利用したいですか【利用希望度】

・作業はどのくらい掛かったと思いますか

・Word を用いて説明書を作成するにあたり，追加してほしい機能があればお聞かせ下さい　（自由記述）

2. PaSTePoST システムについてお聞きします

・完成した文書はあなたにとって満足のできる仕上がりになりましたか【作品の満足度】

・PaSTePoST システムでの画像挿入はしやすかったですか【効率】

・作業は心地よく（不快感なく）できましたか【作業の満足度】

・今後同様の作業をする場合，本システムをどの程度利用したいですか【利用希望度】

・作業はどのくらい掛かったと思いますか

・本システムを用いて説明書を作成するにあたり，追加してほしい機能があればお聞かせ下さい　（自由記述）

・今後，同様に他人に向けた説明書を作成する場合

どのような時 Word を使用し，どのような時 PaSTePoST システムを使用したいですか

（自由記述）

以上.

4.4　結果と考察

本項では 12 人の被験者に対して行った実験の結果について記述する.

本実験では，全部で 4 通りの進め方があったが，それぞれについて結果に大きな変化が出ることは無かったので，数値等はまとめて扱った．実際の作業は画像を扱うのみだったが画像のキャプチャや編集に関して被験者に一任したところ, それぞれ主張する部分や強調する箇所が違い, 方法と手順とが同じ被験者を比べても異なる画像を挿入していた．例えばボタンに着目する文章に対して，ボタンそのものを切り取り表示する手法と，ボタンを含むウィンドウ全体を挿入する手法とがあった．これは本実験において画像そのものに対する制限を設けていなかった為, 被験者それぞれが文章に対し十分と判断した状態の画像になったと考えられる. また事前に用意した文章への不満も見られたが，これは文章の作成を我々が行ったことで手順のまとめ方や分け方を，被験者が分かりにくく感じたためと思われる.

アンケートの結果は, 被験者 12 人の評点の平均を Word と P.P.S で比較したところ，全てにおいて P.P.S が高い評価を得た．各項目の評点の平均と Word と P.P.S との t 検定の結果を表 1 に示す. 有意水準 5%として項目毎に行った t 検定の結果,「効率」($t(11) = 2.20, p = .001$),「作業の満足度」($t(11) = 2.20, p = .032$),「利用希望度」($t(11) = 2.20, p = .046$）の 3 項目で有意差ありの判定を得た．特に表から分かるように「効率」では最も低い値を見ることができ，使いやすさ・作りやすさに関して Word と比較して高い評価を得ることができた．しかし,「作品の満足度」では有意差は認められなかった. これは本実験で行った作業が画像のみに限定されたものであり, 被験者が自ら説明書を作成した，という感覚を持てなかったことが原因と考えられる.

また, 各作業について被験者の体感時間と実際に測定した時間とを比較すると, P.P.S は体感よりも早い時間で作業を終えることができていた. 同一の手順の Word による作成と比べて平均 52%の作業時間で終えることができていた.

「効率」「作業の満足度」の面における有意差および作業時間の短縮が見られたことで，P.P.S の直感的操作によるユーザの作業効率が向上したと考えることができ，本実験で我々が検証したユーザの手間や負担といったものは十分解消できたと言えるだろう．

表 1　各項目に対する方法別の平均点と t 検定の結果

	Word	P.P.S	t 検定における p 値
作品の満足度	3.33	4.00	0.0874
効率	3.00	4.58	0.001024
作業の満足度	3.00	4.00	0.032275
利用希望度	3.42	4.00	0.046342

またアンケートの後半に，被験者に Word と P.P.S で作業を行った上での気づきや追加して欲しい機能について自由に記述してもらった．最も多かった意見として，「キャプラで行う画面のキャプチャ方法が難しかった」,「ウィンドウを複数跨ぐため面倒が増えることがあった」など，Word や P.P.S に対してではなくキャプラの操作や使い勝手に対するものが多く寄せられた．これは被験者全員がキャプラを利用するのが初めてだったため実験前の説明では十分に慣れることができなかったことが考えられる．P.P.S に対しては，画像の編集中マウスをドラッグして範囲を選択する際，選択中のキー操作を忘れ矩形が描かれたことや，矩形の線が跡として残りその後の編集作業に邪魔になるという意見があった．これに対しては再読み込みを行うことで跡は消えたが，恐らくブラウザやコンピュータの処理速度が影響してこのような状態が生まれたのだと推測される．しかし，貼り付け後に編集できる点や貼り付けと保存を同時に行えることについて良かったとした意見もあり，Word と比べるとこういったプラス面の意見が多かった．

ただこれは，日頃から使い慣れたWordと比べ初めて利用するP.P.Sが新鮮で良い部分が多く見られたためであると考えられる．

Word と P.P.S との使い分けを被験者に尋ねたところ，Word は文章に重きを置く，時間をかけて念入りに仕上げるといった場合に利用したいという考えがあった．一方 P.P.S は，シンプルに簡単に作成したい，使用する画像の量が多いといった場合に利用するという意見だった．これに関しては，Word は文章を書くもので P.P.S は画像の挿入と編集を簡単に行うものというイメージが被験者に生まれ，またそれぞれに対する慣れという面もこの問いの答えに影響したのではないかと思われる．

これらを元に考えると，P.P.S ではシステムのユーザビリティという点では手間の解消や使いやすさなど十分に目標を達成していると言える．しかし Word に対して慣れという面で劣っていた為，作業結果への被験者の満足度は低くなってしまったと考えられる．

4.5 今後の課題

アンケートでは Word と P.P.S に追加して欲しい機能について被験者に質問した．

まず Word に対しては，画像への文字や記号の挿入や Word 内での画像編集の機能を追加して欲しいとの意見があった．画像への記号等の挿入に関しては，本実験では P.P.S との比較という面からある程度制限を設けていたが，テキストや記号の挿入は Word 内で可能なため新たに追加する機能とはならない．画像の編集はサイズ変更など数種類しかないが，Word は新しいバージョンが発表される度に機能が増えているため，今後 Word 内で画像の編集を行う方法が追加される可能性がある．

次に P.P.S には Word と同じく画像への文字，記号などの挿入や，編集方法の種類への意見が多かった．編集の種類として，画像の拡大やサイズ変更の際に倍率を細かく指定できる機能，矩形描画に対し線の太さや色を変更できる機能も追加してほしいとの答

えがあった．さらに編集パレットを用いた画像編集や Ctrl キーを用いない右クリックでのコピー&ペースト等も追加機能として挙げられた．全体として P.P.S では画像の編集機能に対する要望が多く，他に文章のフォントや画像位置の調整などへの意見も寄せられていた．編集機能に関しては，今後種類を増やすことで更に多彩な編集を行うことが可能となり，また編集パレットなども既出の研究などを参考に P.P.S ページのサイドバーとして常設することでより編集しやすい環境が整うと思われる．

以上のように様々な追加機能が考えられるため，P.P.S はまだ不完全なシステムであり今後も発展可能なシステムであることが分かった．我々は今回の結果を元に画像の編集方法に限らず画像の貼り付け方法をドラッグ＆ドロップで行うなどの機能も追加し改良を行おうと考えている．

第 5 章　まとめ

5.1　結論

実験において，P.P.S は作業時間の縮小およびユーザビリティの向上を見ることができた. 今回は手順説明書への画像の貼り付けに限り行ったため, その点に関しての P.P.S の有効性は示すことができたと言える．しかし，ユーザの慣れやシステムの機能などにまだ不十分な要素も存在するため, 今後も引き続き新たな機能を追加し更に汎用性を持たせた手法を研究していく必要がある．

本研究では Web ブラウザ上における直感的操作に基づいて，ユーザ自身の感性によって簡単かつ単純な Web ページ構成を行う手法を提案した．Web ブラウザでのコピー&ペースト等の単純操作による構成が，ユーザの負担をどれほど減らすことができるのか, 時間や手間といった作業の効率に関わる問題をどの程度解消できるのかを知るため P.P.S を作成し，対照実験を行うことで比較しその効果を検証した．対照実験の結果から画像の挿入に対しては P.P.S の優位性が認められ，目的としたユーザビリティの向上や手間の解消という観点についても被験者のアンケート結果や自由記述の結果から十分に達成できたと考えられる．

5.2　展望

システムの課題として，画像の編集方法の種類の追加が挙げられる．今回実装していなかった画像の拡大や矩形のスタイルなどを今後追加していく必要があると考えられる．文章についても，実験では事前に準備していたがこれのフォントや大きさも自在に変更できるよう考えなければならない．また，今回数値の面で見た時は満足する結果を得ることができたが, ユーザからの気づきや要望については引き続き十分に考慮して実現していかなければならない．今後，実現可能な追加機能としては，

（1）倍率を指定して画像を拡大／縮小

P.P.S には一定の倍率での画像の縮小機能しか実装していなかった為，今後は細かい単位でのリサイズや拡大機能を考える必要がある．特に拡大に関しては，ある部分に注目させたい場合拡大して見せた方が分かりやすい可能性があるため，必要な機能と言えるだろう．

（2）ドラッグ＆ドロップによる画像の貼り付け／移動

P.P.S ではコピー&ペーストによる貼り付けを行っていたが，これと画像の編集操作において Ctrl キーの使用に戸惑うユーザが見られた．そこでキー操作を極力行わずマウスを中心に行えるようにすると，更に使い易さが増すと思われる．

（3）画像へのテキスト・記号などの挿入

説明書において画像のある部分を強調することは，手順説明では必要なことである．そのため，画像中に例えば矢印や円を描くことでより理解しやすい画像に編集することが可能となる．

（4）編集パレット

前述の通り，キー操作においてたびたび間違いが起きることがあった．そこで切り取りや矩形描画などを 1 つの場所に集めパレットとして利用できるよう実装すると，より単純な操作で画像の編集ができる．

（5）文書の保存

ユーザより P.P.S にて作成した文書も Word と同様に PDF として残したら良いのでは，という意見があった．手元に残すという意味ではそのような方法も好まれるのだろうが，P.P.S では Web ページでの作成という点を踏まえサーバへのアップロードを考えている．ユーザが P.P.S を利用し作成した Web ページをサーバへアップロードし URL を付加する．そしてその URL をユーザが別に持つホームページ等にリンクとして貼り，インターネットを利用する全ての人がアクセス可能になる事を想定している．

P.P.S の優れた点は，Web ブラウザ上で構成を行えるという点である．本稿で軸とした手順説明書を例にとると，通常説明書に添付される画像は，言うなれば静的な画像であるため, その中で強調したい部分がある場合その箇所のみを表示するかマーカーを付けるなどして主張する必要がある．しかし，P.P.S においては Web ブラウザ上で画像編集を行うという利点を生かし，動的な画像へと進化させることができる．例えば画像をクリックするとクリックした部分が拡大表示されるアニメーションの追加, 実際に操作している様子やウィンドウ画面が流れる動画，などが考えられる．こうした機能はユーザの理解をより深め，説明書としての役割を十二分に満たす Web ならではのものと言える．また画像についてだけでなく，今後はテーブルの枠に捉われずユーザの思い思いのページ構成が行える手法を手掛けていく必要がある. そのためマウス位置を認識してアクションを起こしたり，様々な種類のデータやファイルを挿入したり，よりユーザの理想を実現できるページ構成が行えるよう実装・検証を行っていかなければならない.

今後，Web 上のシステムは更に進化していくと考えられる．利用者層もより厚くなり専門的な知識を持たないユーザが Web ページを作成する機会も増えていくだろう．今後の技術の進化に合わせて，我々の提案手法も様々な機能を追加し，よりユーザの視点で“使いやすい”Web ページの構成システムへと進化させていくことが望まれる．その中で簡単かつ単純に Web ページを構成することでユーザの負担を極限まで減らしていくことを想定しながら，本研究を更に進化させていきたい.

謝辞

本研究および論文作成にあたり，終始適切なご指導，ご鞭撻を賜りました三浦元喜准教授に心から御礼申し上げます．また，実験およびアンケートにご協力頂いた九州工業大学工学部の有志の方々に対し心より感謝いたします．誠に有難う御座いました．

参考資料

[1] 総務省　「ブログ・ＳＮＳ（ソーシャルネットワーキングサイト）の現状分析及び将来予測」

http://warp.ndl.go.jp/info:ndljp/pid/283520/www.soumu.go.jp/s-news/2005/050517_3.html

[2] Pasteboard　http://pasteboard.co/

[3] snag.gy　http://snag.gy/

[4] Gyazo　http://gyazo.com/ja

[5] NOTA　http://nota.jp/ja/

[6] qwikWeb　http://qwik.jp/

[7] Ajax　http://ja.wikipedia.org/wiki/Ajax

[8] jQuery　http://ja.wikipedia.org/wiki/JQuery

[9] キャプラ　http://capwrite.softonic.jp/

[10] XAMPP　http://www.apachefriends.org/jp/xampp-windows.html

[11] Google Chrome　http://www.google.co.jp/intl/ja/chrome/

[12] ユーザビリティ

http://ja.wikipedia.org/wiki/%E3%83%A6%E3%83%BC%E3%82%B6%E3%83%93%E3%83%AA%E3%83%86%E3%82%A3

[13] 山田祥寛「10 日でおぼえる jQuery 入門教室」　翔泳社 2011

[14] 江渡浩一郎　「パターン，Wiki，XP - 時を超えた創造の原則」　技術評論社 2009

[15] STRd6 HTML5　「JavaScript Pasting Image Data in Chrome」

http://strd6.com/2011/09/html5-javascript-pasting-image-data-in-chrome/

[16] PHP マニュアル　http://www.php.net/manual/ja/index.php

[17] Baoming Zhang, Man Ren 「Paste image from clipboard using javascript in CKEditor」 ACSA 2012

[18] 杉浦 淳, 小池雄一, 古関義幸 「Internet Scrapbook: Creating Personalized World Wide Web Pages」 ヒューマンインターフェース 1997

付録

付録として，実験において被験者が作成した説明書の例を添付する．

[1] 方法：Word,　　手順：デスクトップの背景の変更

[2] 方法：PaSTePoST システム,　　手順：タイムゾーンの時計の追加